US 20030070119A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2003/0070119 A1

Dallin (43) Pub. Date: Apr. 10, 2003

(54) METHOD AND SYSTEM FOR TESTING A SOFTWARE PRODUCT

(76) Inventor: Michael Dean Dallin, Boulder, CO (US)

Correspondence Address:
HOFFMAN WARNICK & D'ALESSANDRO, LLC
3 E-COMM SQUARE
ALBANY, NY 12207

(21) Appl. No.: 09/973,748

(22) Filed: Oct. 10, 2001

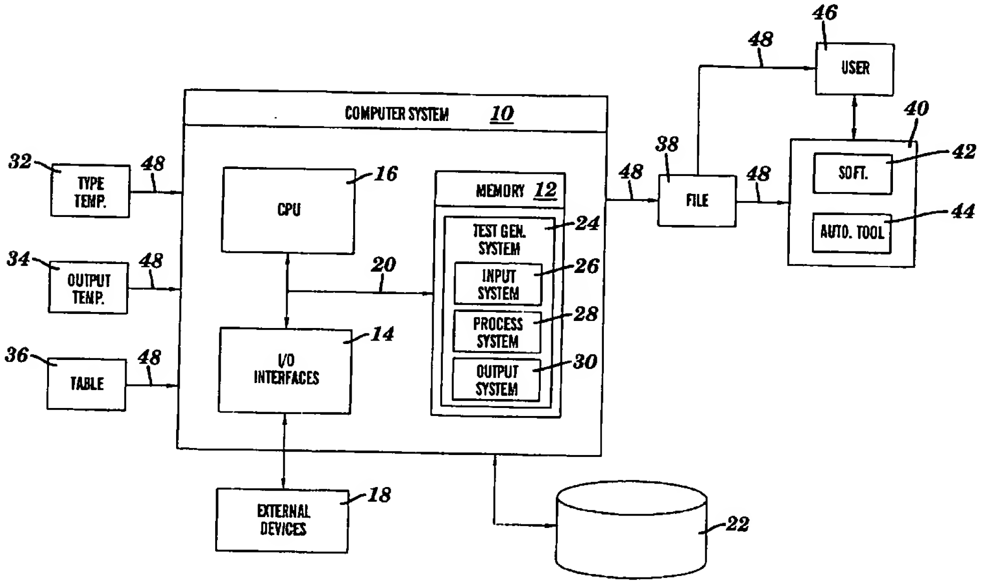
Publication Classification

(51) Int. Cl.^{7} H04L 1/22

(52) U.S. Cl. 714/38

(57) ABSTRACT

A method and system for testing a software product is provided. Specifically, a type template, an output template, and a table of test data pertaining to the software product are provided. The test data is processed by a test generation system based upon the type template and the output template to automatically generate a test script file having at least one test case. The test script file is then used to test the software product.



PGPUB-DOCUMENT-NUMBER: 20030070119

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20030070119 A1

TITLE: Method and system for testing a software product

PUBLICATION-DATE: April 10, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY
RULE-47			
Dallin, Michael Dean	Boulder	CO	US

US-CL-CURRENT: 714/38

ABSTRACT:

A method and system for testing a software product is provided. Specifically, a type **template,** an output **template,** and a table of **test data** pertaining to the software product are provided. The **test data** is processed by a test generation system based upon the type **template** and the output **template** to automatically generate a test script file having at least one test case. The test script file is then used to test the software product.

---------- KWIC ---------

Abstract Paragraph - ABTX (1):

A method and system for testing a software product is provided. Specifically, a type **template,** an output **template,** and a table of **test data** pertaining to the software product are provided. The **test data** is processed by a test generation system based upon the type **template** and the output **template** to automatically generate a test script file having at least one test case. The test script file is then used to test the software product.

Summary of Invention Paragraph - BSTX (10):

[0008] The present invention overcomes the drawbacks of existing systems by providing a method and system for testing a software product. Specifically, the present invention provides a test generation system for automatically generating a test script file having test cases for testing a software product. To generate the test cases, the test generation tool is provided with a type **template,** an output **template,** and a table having **test data** for the software product being tested. The test generation system will process the **test data** based upon the type **template** and generate a test script file in a format dictated by the output **template**. The test script file has the test cases needed to test the software product. Depending on the format set forth in the output **template,** the test script file can be either manually run by a user, or automatically run by an automation tool. In either event, the test script file is run to test the software product.

Summary of Invention Paragraph - BSTX (11):

[0009] According to a first aspect of the present invention, a method for testing a software product is provided. The method comprises the steps of: (1) providing a predefined **template**; (2) providing a table having **test data** for the software; (3) running a test generation system with the **template** to process the

test data and to automatically generate a test script file; and (4) running the software product while using the generated test script file to test the software product.

Summary of Invention Paragraph - BSTX (12):
[0010] According to a second aspect of the present invention, a method for testing a software product is provided. The method comprises the steps of: (1) providing a predefined **template**; (2) providing a table having **test data** for the software product; (3) running a test generation system with the **template** to process the **test data** and to automatically generate an executable test script file; and (4) running the software product while using an automation tool to run the executable test script file to test the software product.

Summary of Invention Paragraph - BSTX (13):
[0011] According to a third aspect of the present invention, a method for testing a software product is provided. The method comprises the steps of: (1) entering a predefined type **template** into a test generation system; (2) entering a predefined output **template** into the test generation system; (3) entering a table having **test data** for the software product into the test generation system; (4) running the test generation system with the type **template** and the output **template** on a first computer system to process the **test data** to automatically generate an executable test script file; and (5) running the software product on a second computer while using an automation tool to run the executable test script file to test the software product.

Summary of Invention Paragraph - BSTX (14):
[0012] According to a fourth aspect of the present invention, a system for testing a software product is provided. The system comprises: (1) an input system for receiving a type **template,** an output **template,** and a table having **test data** for the software product; and (2) a process system for processing the **test data** to automatically generate a test script file based upon the type **template** and the output **template,** wherein the generated test script file is used to test the software product.

Summary of Invention Paragraph - BSTX (15):
[0013] According to a fifth aspect of the present invention, a system for testing a software product is provided. The system comprises: (1) a test generation system stored on a first computer system, wherein the test generation system comprises: (a) means for receiving an output **template,** a type **template,** and a table having **test data** for the software product; (b) means for processing the **test data,** based upon the type **template** and the output **template,** to automatically generate an executable test script file having at least one test case; and (c) means for outputting the executable generated test script file; and (2) an automation tool for running the executable test script file to test the software product, wherein the software product is stored on a second computer system.

Summary of Invention Paragraph - BSTX (16):
[0014] According to a sixth aspect of the present invention, a program product, stored on a recordable medium, for testing a software product is provided. The program product comprises: (1) an input system for receiving a type **template,** an output **template,** and a table having **test data** for the software product; and (2) a process system for processing the **test data** to automatically generate a test script file based upon the type **template** and the output **template,** wherein the generated test script file is used to test the software product.

Detail Description Paragraph - DETX (18):

[0039] In general, the present invention provides a system and method for testing a software product. Specifically, under the present invention a type **template,** an output **template,** and a table of **test data** pertaining to the software product being tested are provided. A test generation system will process the **test data** based upon the type **template** and the output **template** to automatically generate a test script file of test cases. The test script file can then be used to test the software product. In particular, the output **template** dictates the format of the generated test script file. The format can be either a documentation test script file that is manually run by a human user, or an executable file that is automatically run by an automation tool.

Detail Description Paragraph - DETX (23):

[0044] Under the present invention, a type **template** 32, an output **template** 34, and a table 36 having **test data** pertaining to software product 42 are received by input system 26, and optionally stored in database 22. Process system 28 will process the **test data** in table 36 based upon type **template** 32, and automatically generate a test script file 38 having at least one test case. The test script file is generated in a format dictated by output **template** 36. As will be further described below, table 36 includes rows of **test data.** Preferably, a separate table is prepared for each potential use case. For example, if two possible use cases for software product 42 are: (1) test the login dialog; and (2) test the logout dialog, a separate table will be prepared for each use case. Moreover, each use case has one or more test cases. As defined above, a test case is a series of steps required to perform one possible variation of a transaction in a use case. For example, the use case of testing the login dialog could include the test cases of: (1) login with a valid username and password; (2) login with variations of an invalid password; (3) login with blank password; (4) login with blank user name; and (5) test the cancel button. Data for these use cases would be arranged into separate rows of table 36. Process system 28 will process the **test data** based upon type **template** 32 to automatically generate a test script file comprised of the test cases. The test script file will be in a format dictated by output **template** 34. Type **template** 32 and output **template** 34 are preferably flat ASCII files that contain macros and that represent the desired output.

Detail Description Paragraph - DETX (26):

[0047] Cells 50, 52, 54, 55, 56, 56, 57, 58, 59, and 60 set forth the type headings for the cells thereunder. Type headings are user-defined and relate to the object headings set forth in cells 74, 76, 78, 80, and 82. Each type heading is defined in the type **template** and instructs the process system how to generate test cases from the **test data** in table 36, as will be further described below. Object headings define a particular object type in a software product being tested. For example, in an object-oriented GUI in which a user must enter content into a text field, an object heading may indicate a particular text field (e.g., the "user name" text field). Rows 66, 68, 70, 72, and 73 of table 36 each represent separate types of potential test cases. As described above, process system will process the **test data** in table 36 using the type **template** and the output **template** to automatically generate a test script file containing the test cases.

Detail Description Paragraph - DETX (27):

[0048] As shown in cell 62 of table 36, the table is entitled "login" and relates to the use case of testing the login dialog (cell 64). Cells 50, 52, 54, 55, 56, 57 58, 59, and 60 contain the type headings ACTOR, SETUP, TF, TFBLANK, FETF, PB, VWINCLOSES, and VMESSAGE. FIG. 3 depicts a type **template** in which the type headings are defined. Each type heading includes a tag, a description, and a text line that indicates the instruction that will be generated by process system. Moreover, each type heading shown in FIG. 3 utilizes macros (e.g., &lt;cell_contents&gt;). Each macro will effect a copying and pasting of **test data** from the table into the instruction indicated in the text line of each type heading. It should be appreciated that the "instructions" generated can be in a code intended to be implemented by the automation tool or in a text format for a user to manually implement.

Detail Description Paragraph - DETX (28):

[0049] As depicted, type heading ACTOR 90 indicates the actor for each test case, that is, who is performing the test cases. In determining the actor for each test case, the process system will copy the cell contents under the ACTOR type heading for each test case (i.e., row) in the table and paste it into the test script. This functionality is provided by the use of macros. As indicated above, the macros used in type **template** 32 copy and paste **test data** from a particular cell of table 36 to generate the test cases in the test script file. As shown in FIG. 3, for the ACTOR type heading 90, a macro entitled &lt;cell_contents&gt;is utilized. The &lt;cell_contents&gt;macro will cause a copying and pasting of **test data** from a corresponding cell of a test case row. In this case, this will cause the copying and pasting of any cell contents under the ACTOR type heading (column 63) for each test case (i.e., rows 66, 68, 70, 72, and 73) into the test script. Thus, the instruction of "the actor for this case is user" will be generated with respect to the test case in row 66. Moreover, since none of the subsequent rows 68, 70, 72, and 73 change the actor, "user" will remain the actor for the remaining test cases. In a preferred embodiment, a blank cell in a column will have the same **test data** value(s) as the last non-blank cell in the same column. For example, cell 53 will be deemed by process system 28 to have the same actor, namely "user," as cell 51.

Detail Description Paragraph - DETX (35):

[0056] Thus, type **template** 32 provides a key by which test cases are generated based upon the **test data** in table 36. Referring back to table 36, the **test data** is shown. Using the explanation set forth above, the following test cases will be generated:

Detail Description Paragraph - DETX (47):

[0068] It should be understood that the **test data and templates** shown herein are intended to be illustrative only and are not exhaustive. Specifically, **templates** and tables under the present invention are user definable and, accordingly, an infinite quantity of variations could exist. For example, **test data** for one or more additional test cases for testing a login dialog could have been included in table 36. Moreover, it should be appreciated that the present invention is not limited to testing a login dialog.

Claims Text - CLTX (1):

1. A method for testing a software product, comprising the steps of: providing a predefined **template**; providing a table having **test data** for the software product; running a test generation system with the **template** to process the **test data** and to automatically generate a test script file; and running the software product while using the generated test script file to test the software product.

Claims Text - CLTX (2):

2. The method of claim 1, wherein the test script file contains at least one test case generated based upon the **test data and the template**.

Claims Text - CLTX (3):

3. The method of claim 2, wherein the **template** is a type **template** having at least one macro, and wherein the type **template** provides a key for the test

generation system to generate the at least one test case based upon the **test data**.

Claims Text - CLTX (10):
10. A method for testing a software product, comprising the steps of: providing a predefined **template**; providing a table having **test data** for the software product; running a test generation system with the **template** to process the **test data** and to automatically generate an executable test script file; and running the software product while using an automation tool to run the executable test script file to test the software product.

Claims Text - CLTX (11):
11. The method of claim 10, wherein the predefined **template** is a type **template** that provides a key for the test generation system to generate a test case from the **test data**.

Claims Text - CLTX (16):
16. A method for testing a software product, comprising the steps of: entering a predefined type **template** into a test generation system; entering a predefined output **template** into the test generation system; entering a table having **test data** for the software product into the test generation system; running the test generation system with the type **template** and the output **template** on a first computer system to process the **test data** to automatically generate an executable test script file; and running the software product on a second computer while using an automation tool to run the executable test script file to test the software product.

Claims Text - CLTX (17):
17. A system testing a software product, comprising: an input system for receiving a type **template,** an output **template,** and a table having **test data** for the software product; and a process system for processing the **test data** to automatically generate a test script file based upon the type **template** and the output **template,** wherein the generated test script file is used to test the software product.

Claims Text - CLTX (19):
19. The system of claim 17, wherein the type **template** provides a key for generating the test script file based upon the **test data,** and wherein the tests script file includes at least one test case.

Claims Text - CLTX (23):
23. A system for testing a software product, comprising: a test generation system stored on a first computer system, wherein the test generation system comprises: means for receiving an output **template,** a type **template,** and a table having **test data** for the software product; means for processing the **test data,** based upon the type **template** and the output **template,** to automatically generate an executable test script file having at least one test case; means for outputting the executable test script file; and an automation tool for running the executable test script file to test the software product, wherein the software product is stored on a second computer system.

Claims Text - CLTX (24):
24. A program product, stored on a recordable medium, for testing a software product, comprising: an input system for receiving a type **template,** an output **template,** and a table having **test data** for the software product; and a process system for processing the **test data** to automatically generate a test

script file based upon the type **template** and the output **template,** wherein the generated test script file is used to test the software product.

Claims Text - CLTX (26):
26. The program product of claim 24, wherein the type **template** provides a key for generating the test script file based upon the **test data,** and wherein the tests script file includes at least one test case.